UNIVERSITAS SCIENTIARUM SZEGEDIENSIS

UNIVERSITY OF SZEGED

Department of Software Engineering

New Conceptual Coupling and Cohesion Metrics for Object-Oriented Systems

Béla Újházi[1], **Rudolf Ferenc**[1], Denys Poshyvanyk[2] and Tibor Gyimóthy[1]

[1]University of Szeged, Hungary

[2]The College of William and Mary, USA

Background

- Coupling and cohesion measures capture the degree of interaction and relationships among source code elements
- Most coupling and cohesion metrics rely on structural information, which capture relations, such as method calls or attribute usages
- Structural metrics lack the ability to identify conceptual links, which specify implicit relationships encoded in identifiers and comments in the source code

New metrics

- CCBO
  - Conceptual Coupling between Object classes
  - Number of classes which are conceptually related to a class
- CLCOM5
  - Conceptual Lack of Cohesion on Methods
  - Number of strongly connected components in a class when a class is considered as a graph whose vertices are the methods and the edges are the conceptual relations among the methods

Conceptual relations

- The identifiers and comments are parsed and transformed into a corpus of textual documents
    - each document corresponds to the implementation of a method
- LSI (Latent Semantic Indexing) technique takes the corpus and creates a term-by-document matrix
    - captures the dispersion and co-occurrence of terms in class methods
- SVD (Singular Value Decomposition) constructs a subspace, referred to as the LSI subspace
    - methods from the matrix are represented as vectors in the LSI subspace
- The cosine similarity between two vectors is the measure of conceptual similarity between two methods
- Two methods are conceptually related if their conceptual similarity is greater than a given threshold

Advantages

- Simpler to compute
- Basically programming language independent
- CCBO & CLCOM5, when used in conjunction, can predict bugs nearly as precisely as the combination of 58 structural metrics available in the Columbus source code quality framework and can be effectively combined with these metrics to improve bug prediction in Mozilla source code (C++)

Empirical evaluation

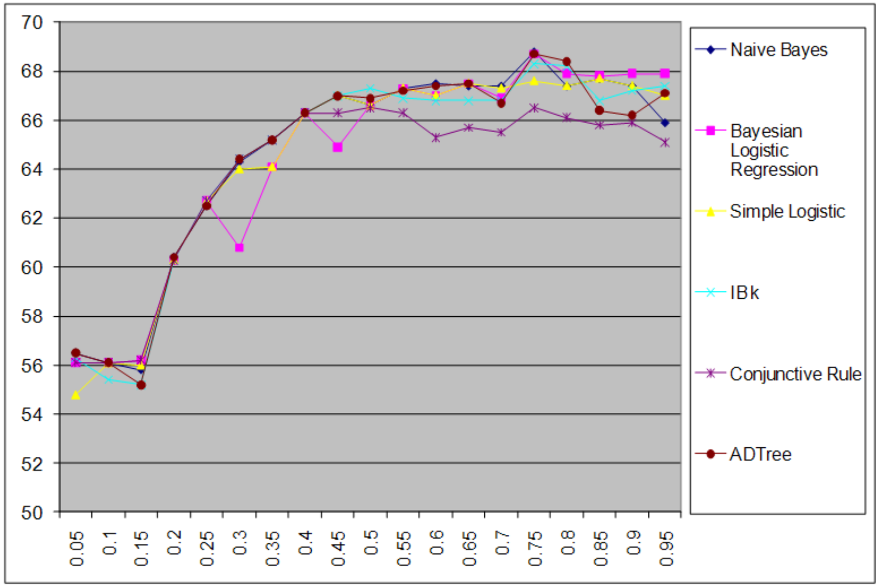
- Research question #1
  - Are the new metrics, CCBO and CLCOM5, orthogonal as compared to existing structural and conceptual coupling and cohesion metrics?
- PCA did not reveal new components because of the new metrics
- Correlation analysis showed that
  - CCBO correlated with CBO and RFC (coefficient between 0.4-0.5)
  - CLCOM5 correlated with LOC, NOI, CBO, RFC, WMC (coefficient above 0.7)
- The new metrics are not othogonal, but are cheaper to compute

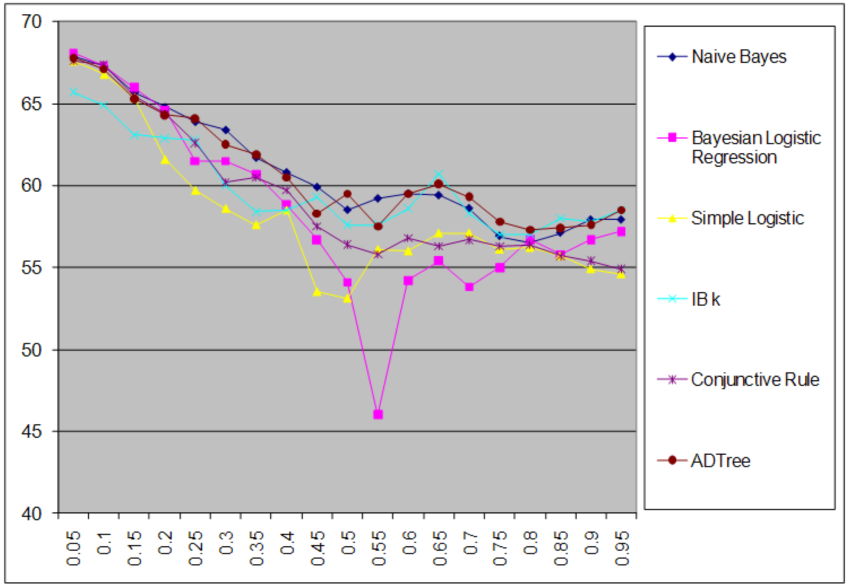
Empirical evaluation

- Research question #2
  - How does stemming impact accuracy of CCBO and CLCOM5 for predicting fault-proneness of classes?
- Result of combining new conceptual metrics (without stemming) for predicting fault-proneness is comparable to the combination of structural metrics
- Models based on conceptual metrics are able to outperform the models based on structural metrics in terms of recall (84.2% vs. 73%) and F-measure (73.9% vs. 72.4%)
- Combining conceptual metrics with stemming and all the structural metrics produces the best values for accuracy (72.7%), precision (82.1%), recall (74.7%) and F-measure (74.2%)
- Stemming improves the results

Empirical evaluation

- Research question #3
  - What is the optimal threshold for CCBO and CLCOM5 for predicting fault-prone classes?





- CLCOM5 peak performance of 68.5% in accuracy is observed in the interval of [0.7, 0.8]
- CCBO best accuracy is observed at threshold 0.05

Empirical evaluation

- Research question #4
  - Does combining CCBO and CLCOM5 with existing structural and conceptual cohesion and coupling metrics improve accuracy of predicting fault-prone classes?
- Combining CCBO and CLCOM5 with structural metrics prediction models are more robust than combinations of existing conceptual metric C3 and structural metrics
  - Accuracy (73% vs. 72.3%)
  - Precision (81.1% vs. 80.9%)
  - Recall (72.9% vs. 73.9%)

Our controversial statement

It is enough to analyze only source code identifiers and comments to reliably measure coupling and cohesion!

Thanks for your attention!